

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

Robocast, Inc.,

Plaintiff,

v.

Civil Action No. 11-235-RGA

Apple Inc.,

Defendant.

MEMORANDUM ORDER

Before the Court is Plaintiff Robocast, Inc.'s Motion for Reconsideration of the Court's April 22, 2014 Summary Judgment Order (D.I. 466), related briefing (D.I. 467), and Defendant Apple's Opposition. (D.I. 477). In granting summary judgment that Top Sites does not infringe, I stated that, "Top Sites does not contain a timing parameter. Robocast does not dispute this, so I accept it as accurate." (D.I. 457 at 9 n. 2) (internal citation omitted). Because I found that Top Sites does not contain a timing parameter, I held that it cannot contain a node, and therefore does not infringe. Robocast asserts that it did in fact dispute the assertion that Top Sites does not contain a timing parameter. Robocast contends that this is a factual oversight and asks that I reconsider the grant of summary judgment that Top Sites does not infringe.

I did overlook the particular argument to which Robocast now cites. The entirety of the argument is one sentence buried in a footnote. It reads, "For Top Sites, global duration values including the *snapshotFreshnessInterval* and *endSafariBusyDelay* properties tell the system how long to display each image on the screen." (D.I. 367 at 14 n. 5). For support, the footnote cites to one page in the Almeroth Infringement Report, which states:

Each Top Site is displayed for a freshness interval defined by *snapshotFreshnessInterval* of 1 hour. “When fetching snapshots automatically, we must never request a snapshot for a URL if a snapshot request for that URL has completed ... within this interval.” *See, e.g.* /Additional_Robocast_20130110/816-WebBrowser-7534.57.2/TopSites/TopSitesController.cpp ln 51-55.

Alternatively, the *endSafariBusyDelay* variable specifies the duration (2 seconds) prior to triggering the snapshot refresh process when Safari has stopped being busy. *See, e.g.* /Additional_Robocast_20130110/816-WebBrowser-7534.57.2/Basics/Application.cpp ln 73.

(D.I. 368-1 at 170).

I do not agree that a conclusory statement buried in a footnote is sufficient to constitute an “overlooked argument.” In the Courts of Appeals, “arguments raised in footnotes are not preserved.” *SmithKline Beecham Corp. v. Apotex Corp.*, 439 F.3d 1312, 1320 (Fed. Cir. 2006). Other District Courts have applied this rule, holding that raising an argument in a footnote was not sufficient to meet that party’s summary judgment burden. *See Morrison v. Wells Fargo Bank, N.A.*, 711 F. Supp. 2d 369, 384 (M.D. Pa. 2010); *Culpepper v. U.S. Fire Ins. Co.*, 2013 WL 2370550, at *4 (N.D. Tex. May 31, 2013).

In any event, even had I considered Robocast’s conclusory statement, I would still have ruled in the same manner. In the summary judgment briefing, Apple’s expert asserted that “there is no timing parameter associated with a [Top Sites] snapshot’s transition from one resource to another resource.” (D.I. 305 at 4, ¶ 10). The timing parameter is “the duration for which the resource’s content is to be presented by default.” (D.I. 239 at 5). Top Sites is a static display. It therefore makes sense that it has no timing parameter. Robocast’s position is that Dr. Almeroth’s infringement report explains how the two pieces of computer code tell the system “how long to display each image on the screen.” The difficulty with Robocast’s assertion that the *snapshotFreshnessInterval* tells the system how long to display each image is that the assertion is not supported by the record.

While Dr. Almeroth concludes that the *snapshotFreshnessInterval* defines the time for which an image is displayed, the code merely states that it will not request a new image if the present image was requested within the previous interval. (D.I. 368-1 at 170). Dr. Almeroth appears to base his conclusion on the deposition testimony of Steven Falkenburg, yet the testimony does not support the conclusion. When asked whether “a user [could] just sit there and...watch the Top Sites wall and consume that content,” Mr. Falkenburg replied that, “if they take an action to show the Top Sites wall, it will...update the snapshots if they’re an hour old, but then they will not change no matter how long you leave it up.” (D.I. 368-1 at 168, quoting Falkenburg Tr. at 201:1-9). It is clear from Dr. Almeroth’s report that the *snapshotFreshnessInterval* does not tell the system how long to display an image.

Apple’s expert, Dr. Menasce, explained how Top Sites functions. In his expert report, Dr. Menasce stated that Snapshots are updated only if they are more than one hour old and the user has performed an action in Safari. (D.I. 305-5 at 57, ¶ 236) (citing the Falkenburg deposition, pp. 19-20). Dr. Menasce explained that the *snapshotFreshnessInterval* is a “constant declared as having the value of one hour.” (D.I. 305-5 at 57, ¶ 237). He explained how Top Sites computes the value of the *expiryTime* variable as the current time minus the *snapshotFreshnessInterval*, and concluded that the one hour *snapshotFreshnessInterval* “does not represent the duration during which a node is to be presented by default... [it] is only used to determine if a snapshot is stale.” (D.I. 305-5 at 58, ¶ 238; 62, ¶ 254). Dr. Menasce opined that the portion of the Falkenburg deposition to which Dr. Almeroth cited supports the conclusion that the *snapshotFreshnessInterval* does not represent the duration during which a snapshot is displayed.

(D.I. 305-5 at 62, ¶ 255). It is clear that *snapshotFreshnessInterval* is not the requisite duration information, and therefore does not form a basis to conclude that Top Sites contains a node.¹

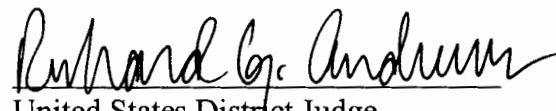
As for the *endSafariBusyDelay*, there is no way this could be the requisite duration information. As explained by Dr. Almeroth, it “specifies the duration prior to triggering the snapshot refresh process when Safari has stopped being busy.” (D.I. 368-1 at 170). This has nothing to do with how long a snapshot is displayed by default. It merely prevents the system from replacing a snapshot when Safari is busy. As Mr. Falkenburg stated, “[i]f that two-second timer expires, then Safari says that...it’s not busy, but the timer is reset every time the user...takes an action that we think will make Safari substantially busy.” (D.I. 368-1 at 170, quoting Falkenburg Tr. at 178:18-22). And as explained by Dr. Menasce, “[t]his value is used as a value of a timer to determine if Safari is busy so that snapshot updates, if they were to occur, will only occur when Safari is deemed to be non-busy. Safari is deemed to be non-busy if 2 seconds have passed without the user taking an action that ‘will make Safari substantially busy.’ If the user takes such an action, ‘the timer is reset.’” (D.I. 305-5 at 63, ¶ 256). Again Dr. Menasce referenced the same Falkenburg deposition, explaining that “the *endSafariBusyDelay* is used to avoid imposing a performance penalty on users who are actively using Safari. So, refreshing of snapshots eligible for refresh keeps being delayed until Safari is deemed not to be busy.” (D.I. 305-5 at 63, ¶ 257) (citing the Falkenburg deposition, p. 155).

Dr. Almeroth’s opinion, and Robocast’s assertion of it as providing evidence that Top Sites has a “duration for which [Top Sites] is to be presented by default,” lack support in the record. They could be properly disregarded as they do not create a dispute of material fact. *See*,

¹ Put another way, as a general matter, Top Sites is of indefinite duration. If a user returns to Top Sites, and the most recent Top Sites images are more than an hour old, they will be updated. There is no particular (or default) time at which the images will be updated.

e.g., *Krippelz v. Ford Motor Co.*, 667 F.3d 1261, 1269 (Fed. Cir. 2012); *Sitrick v. Dreamworks, LLC*, 516 F.3d 993, 1001 (Fed. Cir. 2008) (“Conclusory expert assertions cannot raise triable issues of material fact on summary judgment.”). Plaintiff’s Motion for Reconsideration (D.I. 466) is hereby **DENIED**.

Entered this 11th day of June, 2014.


United States District Judge